

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Тарасова Ирина Владимировна
Должность: Проректор по учебной работе
Дата подписания: 25.05.2022 16:38:16
Уникальный программный ключ:
8c45e14bf77dac42d4f8b124280a05e6949a00d3

**ОБРАЗОВАТЕЛЬНОЕ ЧАСТНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
ПРАВОСЛАВНЫЙ СВЯТО-ТИХОНОВСКИЙ ГУМАНИТАРНЫЙ УНИВЕРСИТЕТ
(ПСТГУ)**

*Факультет информатики и прикладной математики
Кафедра информатики*

**ФОНД
ОЦЕНОЧНЫХ СРЕДСТВ
ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ
ПО ДИСЦИПЛИНЕ**

«Практикум по олимпиадному программированию (часть 1)»

02.03.03 «Математическое обеспечение и администрирование информационных систем»

Профиль:

Администрирование информационных систем

Квалификация выпускника бакалавр

Форма обучения: очная

Москва, 2019 г.

Год начала обучения по учебному плану: 2019

Фонд оценочных средств для текущего контроля успеваемости разработан на основе рабочей программы дисциплины «Практикум по олимпиадному программированию (часть 1)», входящей в состав образовательной программы 02.03.03 «Математическое обеспечение и администрирование информационных систем». Для текущего контроля успеваемости студентов в целях проверки процесса достижения результатов обучения и уровня компетенций, проводится 6 контрольных работ.

Контрольная работа №1

Реализовать пользовательский интерфейс, в котором будут пункты add и tab. При нажатии на пункт add должна появляться форма с полями для ФИО и описания, данные из формы нужно записать в файл in.txt в CSV формате. При нажатии на пункт tab вывести таблицу с данными из файла in.txt. При нажатии на пункт меню «tab» или «add» отправляется GET запрос на сервер, а при нажатии на кнопку «add data» в форме на сервер отправляется POST запрос.

Реализованное приложение должно выглядеть примерно следующим образом:

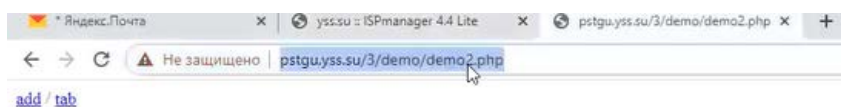


Рисунок 1 – главная страница приложения с меню.

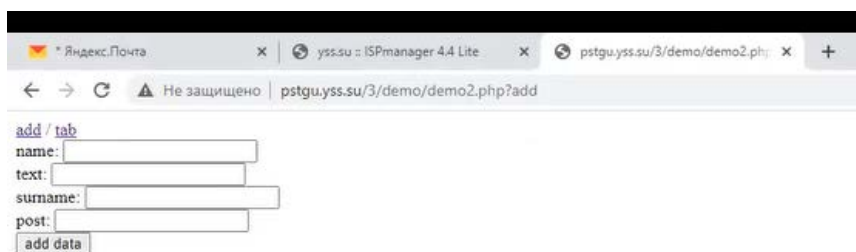


Рисунок 2 – Страница добавления данных.

1	Петров Петрович Петр	бухгалтер
2	Сидор Сидорович Сидоров	менеджер
3	Алексей Алексеев Алексеевич	кладовщик
4	Федор не хочется разговаривать	второкур
5	вапавпв вапвапра вапвап	вапвапвап
6	иван иваныч ивано	директор
7	vfdgdgdfgdf dflhdhfg fghfgh	fgjfgj
1590835722	oj	
1590836075	mn	
1591354731	Пгенчик Хороший Очень	Добавим
1591368624	Почему У меня Не	Получается?
1591464221	45 54 67	123
1611570303	qqq qwqq wqwq	qwqw
1611570323	111 222 333	444

Рисунок 3 – страница вывода данных.

Блок кода на PHP можно отделить от кода на HTML, для этого необходимо обернуть код на PHP в открывающий и закрывающий тег PHP (<?php [код на php] ?>):

```
<?php
header("Content-Type: text/html; charset=utf8");
?>

<a href="lab1.php?add">add</a><a href="lab1.php?tab">tab</a>
```

Для того, чтобы сделать GET запрос на сервер необходимо в атрибуте href тега <a> прописать дополнительные параметры после знака вопроса:

```
<a href="demo.php?add">add</a><a href="demo.php?tab">tab</a>
```

Таким образом, на сервер в скрипт demo.php отправится GET запрос с параметром add или tab.

Проверить получение GET запроса можно следующим образом:

```
if(isset($_GET))
{
    if(isset($_GET['add']))
    {
        // тут обработка запроса на добавление данных в файл
    }
    else if(isset($_GET['tab']))
    {
```

```
// тут обработка запроса на вывод данных их файла
}
}
```

Функция `isset()` возвращает `true` в случае получения GET запроса.

Для того чтобы сделать POST запрос на сервер необходимо в форма `<form></form>` в атрибуте `method` указать метод отправки запроса `post` и в атрибуте `action` указать скрипт, обрабатывающий POST запрос:

```
<form action="demo.php" method="post">
  <p>
    <label>Name</label>
    <input type="text" name="name" required>
  </p>
  <p>
    <label>Description</label>
    <input type="text" name="description" required>
  </p>
  <button type="submit">add data</button>
</form>
```

Проверка получения POST запроса такая же, как для GET запроса:

```
if(isset($_POST))
{
  if( (isset($_POST['name'])) && (isset($_POST['description'])) )
  {
    //тут обработка запроса
  }
}
```

Контрольная работа №2

Реализовать web-интерфейсы для задачи редактирования данных, хранящихся в формате CSV. При этом способ отправки и получения данных должен быть реализован посредством AJAX запросов, можно использовать библиотеку jQuery. Дизайн интерфейсов реализовать на базе CSS стилей, допускается использование библиотеки Bootstrap.

Контрольная работа №3

В продолжении решения контрольной работы №2

1. реализовать скрипт экспорта данных в формат MS Excel;
2. разделить в проекте логику и представление с помощью одного из шаблонизаторов Smarty, Twig, Blade.

Контрольная работа №4

Доработать проект реализованный в контрольной работе №3 таким образом, чтобы редактирование записей в базе данных выполнялось в модальном окне открываемом по клику на соответствующую запись. Данные в модальном окне должны загружаться посредством AJAX запроса.

Контрольная работа №5

Переписать логику модального окна проекта контрольной работы №4 с PHP на Python.

Контрольная работа №6

Продолжая работать над проектом контрольной работы №5 реализовать обмен данными между HTML-таблицей загруженной в браузер и серверным приложением get.php посредством асинхронного обмена по протоколу WebSocket. В работе необходимо создать функцию на JavaScript, которая передает id по onclick и далее id передается по протоколу WebSocket в get.php. На сервере необходимо реализовать соединение с базой данных, и результат ответа обратно вернуть в соответствующее поле таблицы через WebSocket-соединение.

Критерии оценки контрольных работ:

Шкала оценки		Критерии оценки
Оценка	Баллы	
5 (отлично)	9 - 10	Обучающийся: <ul style="list-style-type: none">• полно излагает изученный материал,• дает правильное определение понятий;• обнаруживает понимание материала,• может обосновать свои суждения,

		<ul style="list-style-type: none"> • может привести необходимые примеры не только из учебных пособий, но и самостоятельно составленные; • количество небольших замечаний не более 5; • реализовывает программный код без ошибок компиляции; • программа работает корректно и проходит все тесты;
4 (хорошо)	7 – 8	<p>Обучающийся:</p> <ul style="list-style-type: none"> • полно излагает изученный материал, • дает правильное определение понятий; • обнаруживает понимание материала, • может обосновать свои суждения, • может привести примеры; • количество ошибок не более 5; • реализовывает программный код без ошибок компиляции; • программа работает корректно и проходит все тесты;
3 (удовлетворительно)	5 - 6	<p>Обучающийся:</p> <ul style="list-style-type: none"> • обнаруживает знание и понимание основных положений; • но излагает материал неполно и допускает неточности в определении понятий или формулировках; • не умеет достаточно глубоко и доказательно обосновать свои суждения и привести свои примеры; • количество серьезных ошибок не более 5; • реализовывает программный код без ошибок компиляции; • программа работает корректно и проходит все тесты;
2 (неудовлетворительно)	0 - 4	<p>Обучающийся:</p> <ul style="list-style-type: none"> • обнаруживает незнание большей части соответствующего раздела изучаемого материала, • допускает ошибки в формулировке определений, искажающие их смысл; • количество серьезных ошибок более 5; • нет реализации программного кода либо программа работает не корректно.

Максимальное число баллов, которое может получить студент за работу в течение семестра, равно 80-ти.

Авторы: доц. к.ф.-м..н. Посевин Д.П.

Программа одобрена на заседании кафедры Информатики от «31» мая 2019 года,
протокол № 05-19.